Robust TDOA passive location using interval analysis and contractor programming

Olivier Reynet, Luc Jaulin, Gilles Chabert

Robust TDOA Passive Location Using Interval Analysis and Contractor Programming

REYNET O.
E3I2 Laboratory, EA3876
ENSIETA
Brest, France

JAULIN L.
ENSIETA
Brest, France

CHABERT G.
Contraintes Team
LINA CNRS UMR 6241
Nantes, France

***Abstract*— This paper presents a new approach for solving non-linear passive location problems. It is based on a high level interval modeling language named Quimper. Whereas classical passive location resolutions do not provide any guarantee of convergence to a solution, interval analysis, constraint propagation and contractor programming allow us to avoid any approximations and any linearization. Besides, Quimper naturally provides guarantees on location and bounded error. TDOA passive location configurations are discussed to prove Quimper's efficiency.** *(Abstract)*

Keywords : Passive Location, TDOA, Quimper, Interval Analysis, Constraint Propagation, Contractor Programming, Bounded Error

I. INTRODUCTION

Passive location has been intensively studied in the past years. Numerous devices may actually use passive location including wireless or cellular mobiles [1,2], sonar and radar systems [3], vehicles localization systems. Passive location can take advantage of the growingly number of emitters that are present in environment. It offers a complementary alternative to GPS when indoor, guaranteed or accurate location is required. Moreover, quickly locating an emitter or finding its own location is an important challenge for electronic warfare and unmanned autonomous vehicles (UAV) [4].

Sometimes, decisions have to be taken from results of passive location. When human people are involved, many extra-parameters can be taken into account to choose right actions. For example, symmetrical image location ambiguity in in-line three receivers configuration may be removed from geographical considerations or past observations. In the case of UAVs, there is no human decision maker process. Decisions must be taken from imprecise detectors measurements. That's why exact error estimation is as important as exact location.

This paper intends to show that Quimper software is able to produce an exact bounded error from measurements and, in the same time, to give guarantees on location. Passive location classical approaches give certainly precise results and error estimation. Nevertheless, this error estimation often comes from a probability model which does not care about real-time measured uncertainty. Besides, optimization processes are used to reach a precise result. Theses processes guarantee neither to converge nor to avoid local minimum. That's why an UAV may believe in its location with high level of precision and confidence, while actually being elsewhere. This paper demonstrates the possibility to solve passive location non-linear equations, without optimization, approximations or linearization.

Second section of this paper gives necessary notions about interval analysis, constraint propagation and contractor to understand Quimper software. Third section is dedicated to a simple TDOA passive location configuration which is solved by Quimper's natural contractors. Last section deals with contractor programming concept and is illustrated by an in-line three receivers configuration.

II. INTERVAL ANALYSIS AND CONSTRAINT PROPAGATION

A. *Short Introduction to Interval Analysis*

Assume x is a random variable x of IR. An interval approach introduces [x] to represent an interval $[x_-,x_+]$ which encloses the support of the probability function of x. Standard Operators +, -, ., / and elementary functions like exp, log, sin or cos can be defined and applied on intervals.

To understand basic idea of interval analysis, consider two variables x and y. Assume that they belong to some prior intervals $[x_-,x_+]$ and $[y_-,y_+]$. Addition and difference of intervals can be defined as follow:

$$[x_-, x_+] + [y_-, y_+] = [x_- + y_-, x_+ + y_+] \tag{1}$$

and

$$[x_-, x_+] - [y_-, y_+] = [x_- + y_+, x_+ + y_-] \tag{2}$$

Multiplication and inversion of intervals are:

$$[x_-,x_+]*[y_-,y_+]=[\min(x_-y_-,x_-y_+,x_+y_-,x_+y_+),\\ \max(x_-y_-,x_-y_+,x_+y_-,x_+y_+)] \quad (3)$$

and

$$\frac{1}{[x_-,x_+]}=\begin{cases} IR & \text{if } 0\in[x_-,x_+] \\ \left[\frac{1}{x_+},\frac{1}{x_-}\right] & \text{otherwise.} \end{cases} \quad (4)$$

Let's give some numerical examples of interval computations:

Addition: $[1,5]+[-3,8]=[-2,13]$

Difference: $[1,5]-[-3,8]=[-7,8]$

Multiplication: $[1,5]*[-3,8]=[-15,40]$

Inversion: $\frac{1}{[3,8]}=\left[\frac{1}{8},\frac{1}{3}\right]$

Interval arithmetic tools have been developed[1] and it is now possible and simple to build robust programs directly handling uncertainties [5].

B. *Interval Constraint Propagation*

Interval constraint propagation combines interval computation [6] and constraint propagation [7]. The combination of these two tools has been first presented independently by Clearly [8] and Davis [9]. Constraints can be used to contract the prior feasible domains by removing inconsistent values in the domains of the variables.

For example, consider the three variables x,y, and z and assume they belong to prior feasible domains as follow:

$$\begin{cases} x\in[1,5] \\ y\in[-3,8] \\ z\in[-5,2] \end{cases} \quad (5)$$

Assume also that these three variables are linked by the ADD constraint:

$$z=x+y \quad (6)$$

Then, feasible domains for the variables can easily be contracted by taking into account (6). As this constraint is a primitive constraint, a simple projection algorithm exists:

Algorithm ADD_PROJECTION(inout: [z],[x],[y])

$$[z]:=[z]\cap([x]+[y])$$
$$[x]:=[x]\cap([z]-[y])$$
$$[y]:=[y]\cap([z]-[x])$$

Our numerical example leads to:

$$[x]=[1,5]$$
$$[y]=[-3,1]$$
$$[z]=[-2,2]$$

[y] and [z] have been reduced following ADD constraint. No solution has been removed from these intervals.

Projection is a simple way to contract initial domains. In this case, it is possible because the constraint addition is a trivial constraint. Fortunately, more complex propagation processes have been found. If constraints like equations or inequalities exist between the random variables, a constraint propagation process can produce efficient polynomial algorithms to solve constraints and compute precise solution intervals [5]. Besides, this process guarantees that these computed intervals enclose all solutions for given initial intervals.

There are many algorithms to generate constraint propagation. Nevertheless, one of the most efficient way is forward and backward algorithm [10]. Quimper Software implements forward and backward algorithm. But, in addition, it focuses on another concept: contractors [11].

C. *Quimper Software and Natural Contractor*

Quimper Software is a recent high-level language for QUick Interval Modeling and Programming in a bounded-ERror context (QUIMPER). It is based on Profil/BIAS[2] and IBEX libraries. It has been created to democratize contractor programming, constraint propagation process and interval analysis. Even if Quimper syntax is simple, manual and examples are available on Quimper's web site[3].

Quimper's paver algorithm is a generic solver. It takes a list of contractor, an initial box and follows a classical recursion: the contractors are successively called on the current box until either it gets empty or no more contraction could be done. In the latter case, the box is bisected and contractors are called back again.

Contractor's definition is given in [11]. Thinking of contractor programming as an extension of constraint programming is valid to the extent that contractors help in modelling the output of a problem. But, fundamentally, there is not such an extension since constraints basically tell the "what" whereas contractors tell the "how".

If no special contractor is given, then Quimper uses natural contractors which result from constraints broken into several trivial constraints. Forward-backward propagation is then applied by contractor to reduce boxes. Nevertheless, Quimper presents specific operators to program ad-hoc contractors. A solver can then be programmed, rather than configured, by combining different contractors. These new contractors faster reduce initial domains. But let first try to solve TDOA hyperbolic equations using Quimper language.

[1] See for example INRIA COPRIN project. An in-line interval analysis solver is available at http://www-sop.inria.fr/coprin/ael/form.html.

[2] See Profil/BIAS web site at http://www.ti3.tu-harburg.de/Software/PROFILEnglisch.html

[3] See IBEX QUIMPER site at http://ibex-lib.org/

III. TDOA PASSIVE LOCATION WITH QUIMPER SOFTWARE

TDOA passive location requires two stages: the first stage requires TDOA estimation for each pair receiver-receiver. The second stage uses TDOA estimation to build a set of non-linear hyperbolic equations. In this paper, we suppose that the first stage, TDOA estimation, is already done, for example by correlation techniques [12,18].

A. *TDOA Hyperbolic Equations*

Consider the classical TDOA passive location configuration composed with one emitter and three receivers as sketched on figure 1. Let (x,y) be the location of the emitter, and (x_i,y_i) the location of the receivers. Distance from emitter to receiver i is:

$$D_i = \sqrt{(x-x_i)^2+(y-y_i)^2} \tag{7}$$

If $d_{ij} = D_i - D_j$, we have:

$$d_{ij} = \sqrt{(x-x_i)^2+(y-y_i)^2} - \sqrt{(x-x_j)^2+(y-y_j)^2} \tag{8}$$

Let t_{ij} be the measured Time Difference Of Arrival (TDOA) of the signal between receiver i and j. As $d_{ij} = ct_{ij}$, hyperbolic TDOA equations are:

$$ct_{ij} = \sqrt{(x-x_i)^2+(y-y_i)^2} - \sqrt{(x-x_j)^2+(y-y_j)^2} \tag{9}$$

where c is the speed of the signal and $(i,j) = \{(0,1),(1,2),(2,0)\}$.

Solving these non-linear equations is not a trivial problem. Closed solutions of these equations are known [13,14]. However, these closed forms do not provide any error estimation. Others accurate approaches linearize this set of equations through second order Taylor-series expansion [15,16,17], but need a good initial guess and intensive computation. Our approach based on interval analysis, constraint propagation and contractor programming allow us to avoid any approximations and give true error estimation. So, let us try to model TDOA passive location with Quimper.

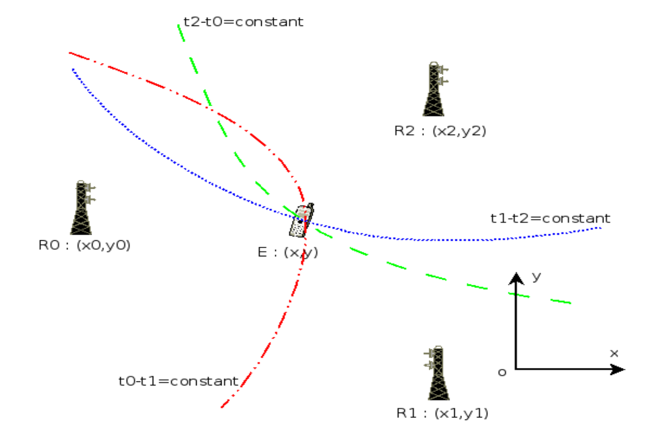

Figure 1. TDOA classical configuration : three receivers and one emitter. Emitter is located at the intersection of the three hyperbolas for which t_i-t_j is a constant.

B. *Quimper Example of TDOA Passive Location*

Unknown variables of this problem are x and y, or, more exactly, [x] and [y]. Indeed, more than an accurate result which might be obtained by others approaches, we are looking for domains which might contain solutions and domains which do not contain solutions. Suppose that the bounded receiver sensibility we are using allows us to define some prior feasible domains for these two variables. Let [x] and [y] belong to [-10000,10000] m, which corresponds to a 400 km^2 initial searching area, about four times Paris urban area. Suppose that transmission medium is made of air and that electromagnetic waves are used. The signal speed c is then 3e8 $km.s^{-1}$. The three receivers are located at R_0 (-9000, 0) m, R_1 (9000,-9000) m and R_2 (9000,9000) m. Emitter is located at (1540,345) m.

In this simulation, measurement errors are introduced by specifying intervals for t_{ij}, instead of a simple real value. We first extract the exact values of time of arrival from geometrical knowledge. Then, fixed or random values may be added or subtracted to exact t_{ij} value to generate errors. In this first example, we supposed that t_{ij} is known with an uncertainty of plus or minus τ = 500 ns. Initial domain of t_{ij} is then 1 µs wide. This uncertainty corresponds to an analog to digital converter with bad precision and a basic signal correlation. Therefore, ct_{ij} belong to domain [$c(t_{ij}- \tau)$, $c(t_{ij}+ \tau)$]. We can now write the following Quimper input file (see Script I). A Quimper file is divided in several parts, including constants and variables declarations, and contractors list. It should be noticed that constants may be intervals but that variables must be an intervals.

SCRIPT I: TDOA AND NATURAL CONTRACTOR

```
Constants
   x0=-9000.0;
   y0=0.0;
   x1=9000.0;
   y1=-9000.0;
   x2=9000.0;
   y2=9000.0;
   ct01 in [-1561.80577055,-1261.80577055];
   ct12 in [381.137771356,681.137771356];
   ct20 in [730.667999191,1030.66799919];
Variables
   x in [-10000,10000];
   y in [-10000,10000];
contractor hyperbola1
   sqrt((x-x0)^2+(y-y0)^2)-sqrt((x-x1)^2+(y-y1)^2) in ct01;
end
contractor hyperbola2
   sqrt((x-x1)^2+(y-y1)^2)-sqrt((x-x2)^2+(y-y2)^2) in ct12;
end
contractor hyperbola3
   sqrt((x-x2)^2+(y-y2)^2)-sqrt((x-x0)^2+(y-y0)^2) in ct20;
end
contractor isThick
   maxdiamGT(10)
end
```

At the end of this Quimper's file are listed all contractors. Here, we simply put hyperbolic equations: Quimper builds natural contractors corresponding to these constraints. Each contractor removes boxes that are not feasible, one after another. Then, if no more domain contraction is observed, domains are bisected and contractors applied on these new domains. Last contractor is a special one called "Thickness contractor" which uses built-in operator *maxdiamGT*: *isThick* removes all boxes that are not thick enough. It is useful if we want solutions to be wrapped into intervals with a size lower than a limit. In this example, this lower size limit is 10m. It means that boxes which contain solutions are contracted by *isThick* and have a maximum dimension of 10m.

C. TDOA Passive Location Results

Figure 2 shows results extracted from Quimper. Time calculation is about 0.088 s on a Intel Core Duo at 2GHz. Boxes removed by hyperbolas contractors are painted in black and grey. Indiscernible boxes removed by *isThick* which may contain solutions appear in white, in the center of the figure near exact emitter location (1540,345).

It can be seen that wide domains are removed with only one operation: if a domain do not belong to any hyperbola, then it is no use to search in. The corresponding contractor removes it. All in all, 1745 boxes have been created to solve this problem.

Initial searching area is 400 km². From this area, *hyperbolas* contractors remove 99.990 % and guarantee that there is no solutions in. *isThick* removes 0.001% of this area which may contain solutions and represents about 37834 m². This result tells that the emitter is located in a 195x195 m² area centred in (1540,342) m. Concretely, it means that we have located somebody in a big stadium, whereas we were looking for him on an area four times greater than Paris.

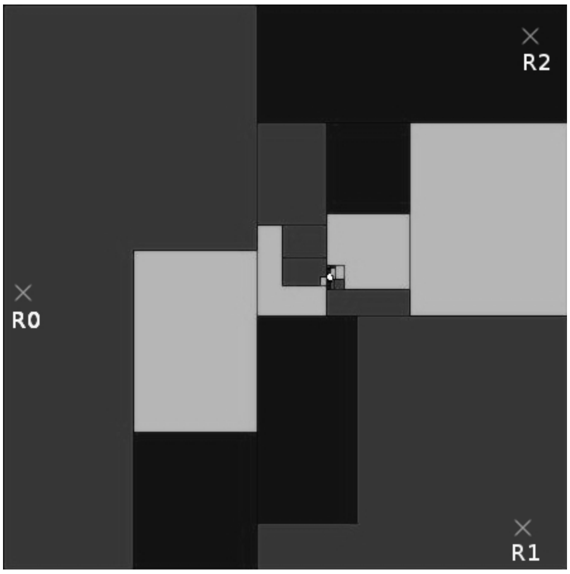

Figure 2. Solution of TDOA equations with natural contractors: removed boxes from *hyperbola1* (dark grey), *hyperbola2* (black), *hyperbolas3* (light grey) and *isThick* (white). Receivers are sketched with crosses: R_0 (-9000, 0) m, R_1 (9000,-9000) m and R_2 (9000,9000) m. Emitter is located at (1540,345) m.

D. Discussion

These results are fairly remarkable for several reasons: first, no approximations are made, though non-linear equations are quickly solved. These computations may be further accelerated, since parallel-computing may be used by interval analysis. Secondly, removed boxes are guaranteed without any solution: no global minimum has been missed because of a local one and no initial guess is needed. Thirdly, error estimation naturally results from computation and may come from real-time error measurements. If an *isThick maxdiamGT* parameter of 1 m is chosen instead of 10 m, *isThick* area is computed at 35097 m² instead of 37834 m². It means that no time consuming small bisections are needed to reach precise results.

Time uncertainty can be tuned. If τ equals 50 ns instead of 500 ns and *maxdiamGT* parameter remains 10 m, then *isThick* area is 671 m², which represents 0.0002% of the initial are. Location area is now reduced to a big tennis court centred at (1538,344). *isThick* area can hence be seen as a true characterization of time measurement error.

IV. TDOA and Contractor Programming

A. Constraint List and Built-In Contractors

Quimper's key idea is that a solver can be programmed rather than configured by combining different contractors. The previous example shows that natural contractors can work concurrently. Let's now design new contractors via Quimper language.

Let's try to compute again the three receivers and one emitter configuration of section III.B, but in a more efficient way (see Script II). Constants and variables are the same as in first script. However, instead of using natural contractors built by Quimper, three stages are proposed to build better ad-hoc contractors.

Script II: TDOA and Contractor Programming

```
Constants
   x0=-9000.0;
   y0=0.0;
   x1=9000.0;
   y1=-9000.0;
   x2=9000.0;
   y2=9000.0;
   ct01 in [-1561.80577055,-1261.80577055];
   ct12 in [381.137771356,681.137771356];
   ct20 in [730.667999191,1030.66799919];
Variables
   x in [-10000,10000];
   y in [-10000,10000];
constraint-list hyperbolas
   sqrt((x-x0)^2+(y-y0)^2)-sqrt((x-x1)^2+(y-y1)^2) in ct01;
   sqrt((x-x1)^2+(y-y1)^2)-sqrt((x-x2)^2+(y-y2)^2) in ct12;
   sqrt((x-x2)^2+(y-y2)^2)-sqrt((x-x0)^2+(y-y0)^2) in ct20;
end
contractor-list clhyp
   inter i=1:3;
      hyperbolas(i)
   end
end
contractor propInter
   propag(clhyp)
end
contractor isThick
   maxdiamGT(10)
end
```

First stage gives a constraint list which takes into account all links between variables. Here, only three equations are used, but others equations may easily be added in this list. For example, equations from others redundant receivers, phase interferometers or radiogoniometers can be introduced in terms of (x,y). No instruction change is needed to handle inconsistent equations system: adding equation to the list is sufficient. Moreover, the greater the number of different constraints is, the faster boxes are removed from initial domain.

Second stage specifies that boxes which do not satisfy intersection of the three constraints can not be solution of our problem. This stage is Quimper's translation of geometrical intersection of the three TDOA hyperbolas. It is done thanks to *inter* built-in contractor. In the script, *hyperbolas(i)* is implicitly interpreted as the natural contractor associated to the i^{th} equation. Hence, this stage creates a new contractor *clhyp* which is the intersection of the three natural contractors.

First script's strategy applies natural contractor only one time and one after another. However, repeating several times the same contractor can contract domains in a more efficient way. Therefore, third stage uses the built-in contractor *propag* to repeat the intersection contractor until no more contraction is observed. Then, *isThick* contractor is applied. As a last resort, indiscernible domains are bisected and entire process restart on these new domains.

B. Contractor Programming Results

Figure 3 shows computation results of Quimper for the prior example, but with contractor programming as described in Script II. *propInter* contractor removes boxes and leads to the same result, but has created fewer boxes (1207 instead of 1217). Computation time is about the same than with natural contractors (0.088 s). It means that formula (9) provides a way to build an efficient natural contractor to solve TDOA equations.

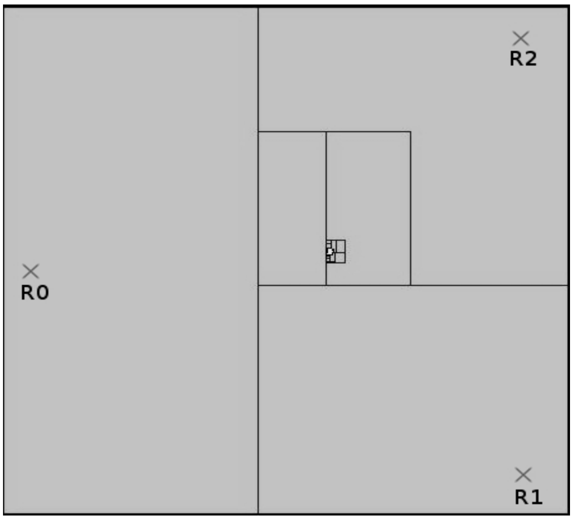

Figure 3. Solution of TDOA equations with programmed contractors: removed boxes *propInter* (light grey) and *isThick* (white). Receivers are sketched with crosses: R_0 (-9000, 0) m, R_1 (9000,-9000) m and R_2 (9000,9000) m. Emitter is located at (1540,345) m

C. In-Line TDOA Passive Location Configuration

Consider now the following in-line configuration: emitter is at (8000,7000) m and receivers are located at R_0 (-9000,0) m, R_1 (0,0) m and R_2 (9000,0) m. This configuration is a bad one, because in-line receiver x-location generates y-ambiguity: two symmetrical solutions are indiscernible. Quimper's results are shown figure 4. *isThick maxdiamGT* parameter is 10 m. Time measurement uncertainty τ is 500 ns.

This example demonstrates that interval analysis and contractor programming do not remove any solution. In a pedagogical way, it also shows that this configuration generates bigger error location: initial area is 400 km² and *isThick* area is about 260557 m² (i.e. 0.06 %). Reasons are time uncertainty, receivers' proximity and in-line topology.

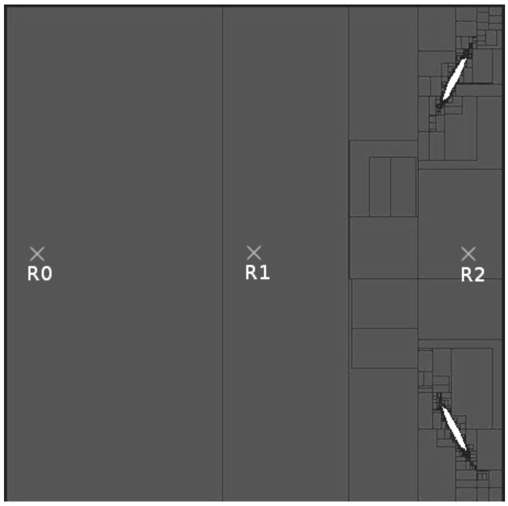

Figure 4. Solution of TDOA equations with programmed contractors: removed boxes from *propInter* (dark grey) and *isThick* (white). Receivers are sketched with crosses: R_0 (-9000, 0) m, R_1 (0,0) m and R_2 (9000,0) m. Emitter is located at (8000,7000) m

V. CONCLUSION

Quimper is a new powerful tool. To our mind, it is the first time that passive location non-linear equations as hyperbolic TDOA equations are solved without any approximations thanks to interval analysis and contractor programming. Bounded-error estimation naturally results from Quimper's computation. Contractor programming is a way to create ad-hoc and more efficient contractors. Finally, this high level interval modeling language provides means to mix different passive location approaches. Hyperbolic TDOA, goniometry or phase interferometry may easily be introduced in the Quimper's simple formalism.

REFERENCES

[1] An Enhanced Two-Step Least Squared Approach for TDOA/AOA Wireless Location. Anchorage, Alaska, USA: ICC 2003, MAY 2003.

[2] L. Cong and W. Zhuang, “Nonline-Of-Sight Error Mitigation In Mobile Location,” Wireless Communications, IEEE Transactions on, Vol. 4, No. 2, pp. 560–573, 2005.

[3] Z. Jiabing, T. Liang and H. Yi, “Study on Moving Target Detection to Passive Radar Based on FM Broadcast Transmitter,” J. of Systems Engineering and Electronics, Vol. 18, No. 3, pp.462-468, 2007.

[4] S. Drake, K. Brown, J. Fazackerley and A. Finn, “),” Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005. Proceedings of the 2005 International Conference on. pp.403- 409, 5-8 Dec., 2005.

[5] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics, Springer, Ed., 2001.

[6] R. E. Moore, Interval Analysis. Englewood Cliffs, NJ: Prentice-Hall, 1966.

[7] A. K. Mackworth, “Consistency in Networks of Relations,” Artificial Intelligence, VOL. 8, NO. 1, PP. 99–118, 1977.

[8] J. G. Cleary, “Logical arithmetic,” Future Computing Systems, vol. 2, no. 2, pp. 125–149, 1987.

[9] E. Davis, “Constraint propagation with interval labels,” Artificial Intelligence, vol. 32, no. 3, pp. 281–331, 1987.

[10] L. Jaulin, I. Braems, M. Kieffer, and E. Walter, Scientific Computing, Validated Numerics, Interval Methods, Proceedings of SCAN 2000. Kluwer Academic Publishers, 2001, Ch. Nonlinear State Estimation Using Forward-Backward Propagation Of Intervals, pp. 191–204.

[11] G. Chabert and L. Jaulin, “QUIMPER, A Language For Quick Interval Modeling And Programming In A Bounded-Error Context,” Artificial Intelligence, 2008.

[12] G. Carter, “Time delay estimation for passive sonar signal processing,” IEEE Trans. on acoust., speech, and signal processing, vol. ASSP-29, no. 3, pp. 463–470, June 1981.

[13] B. T. Fang, “Simple Solutions for Hyperbolic and Related Position Fixes,” Aerospace and Electronic Systems, IEEE Transactions on, VOL. 26, NO. 5, PP. 748–753, 1990.

[14] Y. T. Chan and K. C. Ho, “A Simple and Efficient Estimator for Hyperbolic Location,” Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on], Vol. 42, No. 8, pp. 1905–1915, 1994.

[15] W. Foy, “Postion-Location Solutions by Taylor-Series Estimation,” IEEE Transactions on Aerospace Electronic Systems, Vol. 12, pp. 187–194, March 1976.

[16] D. Torrieri, “Statistical Theory of Passive Location Systems,” IEEE Transactions on Aerospace Electronic Systems, Vol. 20, pp. 183–198, March 1984.

[17] J. Abel, “A Divide and Conquer Approach to Least-Squares Estimation,” Aerospace and Electronic Systems, IEEE Transactions on, Vol. 26, No. 2, pp. 423–427, March 1990.

[18] C. Knapp and G. Carter, “The Generalized Correlation Method for Estimation of Time Delay,” Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on, VOL. 24, NO. 4, PP. 320–327, 1976